2-1-1996

The Investigation of an implementation of SGML based publishing of an graduate thesis

Jin Park

The Investigation of an Implementation of SGML Based Publishing of a Graduate Thesis

by

Jin Park

A thesis project submitted in partial fulfillment of the requirements for the degree of Master of Science in the School of Printing Management and Sciences in the College of Imaging Arts and Sciences of the Rochester Institute of Technology

February 1996

Thesis Advisor: Professor Frank Cost

School of Printing Management and Sciences
Rochester Institute of Technology
Rochester, New York

Certificate of Approval

Master's Thesis

This is to certify that the Master's Thesis of

Jin Park

With a major in Graphic Arts Publishing
has been approved by the Thesis Committee as satisfactory
for the thesis requirement for the Master of Science degree
at the convocation of

February 1996

Thesis Committee:

Frank Cost
Thesis Advisor

Marie Freckleton
Graduate Program Coordinator

Director or Designate

Title of Thesis: **The Investigation of an Implementation of SGML Based Publishing of a Graduate Thesis**

I, Jin Park, hereby, **grant permission** to the Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Date: February 1996

Acknowledgments

I wish to thank Brian Travis and Lori Defurio of Information Architects, Inc., and Dale Waldt of the Research Institute of America. Their combined knowledge, experience, and willingness to share their resources and time are much appreciated. I must also thank Ryan M. O'Grady for his support throughout my study, but especially during times of unexpected tribulation.

Table of Contents

List of Figures

Abstract

The Standard Generalized Markup Language (SGML) has been the International Organization of Standardization (ISO) published standard for text interchange for nearly a decade. Since 1986, SGML based publishing has been successfully implemented in many fields, notably those industries with massive and mission-critical publishing operations such as the military, legal, medical, and heavy industries.

SGML based publishing differs from the WYSIWYG paradigm of desktop publishing in that an SGML document contains descriptive, structural markup rather than specific formatting markup. Specific markup describes the appearance of a document and is usually a proprietary code which makes the document difficult to re-use or interchange to different systems.

The structurally generic markup codes in an SGML document allow the fullest exploitation of the information. An SGML document exhibits more re-usability than a document created and stored in a proprietary formatting code. In many cases, workflow and production are greatly improved by the implementation of SGML based publishing. Historical and anecdotal case studies of many applications clearly delineate the benefits of an SGML based publishing system. And certainly, the boom in Web publishing has spurred interest in enabling a publishing system with multi-output functionality. However,

implementation is associated with high costs. The acquisition of new tools and new skills is a costly investment. A careful cost-benefit analysis must determine that the current publishing needs would be satisfied by moving to SGML. Increased productivity is the measure by which SGML is adopted.

The purpose of this thesis project is to investigate the relative benefits and requirements of a simple SGML based publishing implementation. The graduate thesis for most of the School of Printing Management and Sciences at the Rochester Institute of Technology was used as an example.

The author has expanded the requirements for the publication process of a graduate thesis with factors which do not exist in reality. The required output has been expanded from mere print output to include publishing on the World Wide Web (WWW) in the Hypertext Markup Language (HTML), and to some proprietary electronic browser such as Folio Views for inclusion in a searchable collection of graduate theses on CD-ROM. A proposed set of tools and methods are discussed in order to clarify the requirements of such an SGML implementation.

Chapter 1

Introduction

The Standard Generalized Markup Language (SGML) is nearing a decade in existence as an official standard for document interchange. In 1986, The International Organization for Standardization (ISO) accepted and published the SGML specifications known as International Standard 8879. The implementation of SGML based publishing in diverse industries has proven it to be a beneficial way of handling data in the era of electronic publishing. *SGML and Related Standards* and *The SGML Implementation Guide* are tradebooks which contain case studies of the successful implementation of SGML in various industries.

Electronic publishing systems enable the creation, management, and integration of textual, graphical and audio information for a wide array of output options. The problem with these systems is that the published information is maintained in deliverable formats which are not often interchangeable to different systems. Difficulties also arise within a closed system when information requires modification or alternate output paths.

SGML is a meta-language for defining a descriptive document markup language that enables publishers to take full advantage of the power and speed offered by electronic

publishing systems today. This allows the fullest exploitation of the information. However, the benefits of adopting SGML must be weighed against rather heavy implementation issues. Increased productivity comes with the costs associated with a change in publishing systems.

The purpose of this thesis project is to investigate the relative benefits of and the requirements for an implementation of an SGML based publishing system. The publication of a graduate thesis in the RIT School of Printing Management and Sciences will be used as an example. A set of tools and methods will be proposed and compared to the current publishing methods.

Publishers, authors, and editors will be interested in how SGML might benefit their operations and provide insight into how to best meet their publishing requirements. This thesis project will be evaluated by its ability to illuminate important SGML implementation issues and the relative benefits of such a system.

Chapter 2

A Technical Understanding of SGML

It is not the purpose of this thesis to survey every technical nuance of the Standard Generalized Markup Language. The tradebooks discussed in the Literature Review section of this chapter provide detailed documentation to the ISO standard. *The SGML Handbook* by Charles Goldfarb is the complete text of ISO 8879 along with the author's own annotations. However, understanding an SGML construct is vital to beginning an implementation. This chapter will describe the basic components of an SGML document.

2.1. An SGML document

An SGML document is composed of three parts: the SGML declaration, the document type definition (DTD), and the document instance.

2.1.1. The SGML Declaration

The SGML declaration defines the syntax of SGML markup and the character set used within the document. It precedes the DTD and in its absence, most SGML parsers will choose a default called the *reference concrete syntax*. High levels of customization are

allowed in the SGML declaration. If the system required it, the reference concrete syntax can be customized to a *variant concrete syntax*.

Most implementations require a minimum level of variations from the default settings. An example of a common tailoring of an SGML declaration is in the quantity set portion. These set limits on markup constructs such as the number of attribute names in an attribute definition list, or the length of an element name. For example, the default name length of an SGML markup construct is eight characters. It is often difficult to read the meaning of an object name that has been shortened to eight characters. Yet, to increase the name length can slow down productivity when SGML documents are tagged by hand. In a situation where full or partial auto-tagging will be used, clearer and longer names may be considered more important.

A typical SGML declaration is shown in Appendix A. The *namelen* value has been set to 32 characters thus making this declaration a *variant concrete syntax*.

2.1.2. The Document Type Definition

The Document Type Definition (DTD) defines the rules for marking up a class of documents. A thorough document analysis is one of the first steps in an SGML implementation, and the DTD is developed from this initial analysis. The DTD reflects the purpose of the document and establishes the groundwork for accomplishing the publishing goals. It is not uncommon for a DTD to undergo several revisions as the documents and the goals of the publishing system evolve.

Every piece of a document's structure (as determined by a thorough document analysis) is expressed in the DTD as an element, attribute, or entity. These are called markup declarations. The structure of a document is expressed hierarchically and is controlled by syntactical rules defined by glyph delimiters defined in the SGML declaration. These rules control the allowed usage for elements in the document instance such as order, frequency, tag minimization, and more. A DTD for a graduate thesis was written by the author as part of the document analysis for this thesis project. The complete thesis DTD can be seen in Appendix B. Figure 5 illustrates a tree diagram of the thesis structure. The tool used to generate this diagram is called Near & Far Lite.[1]

2.1.2.1. The Element Declaration

An element is a node on the document structure tree. Hierarchy is expressed by the content model of an element. For example, the highest level structure of a graduate thesis is a thesis. The element thesis contains: front matter, body matter, and back matter. The body matter may contain chapters, and chapters contain a repeatable sequence of sections or paragraphs and sections. Sections contain a title, paragraphs, and more sections (subsections). The lowest level of an element is expressed by declaring the element content. The declared content of a title and a paragraph is actual text, so the declared content of the element title is *#PCDATA*.

Following is a simplified excerpt of the thesis DTD to illustrate the previous narrative description of the content model of a thesis:

```
<!ELEMENT thesis     - - (front, body, back)          >
<!ELEMENT body       - - (chapter)+                   >
<!ELEMENT chapter    - - (title, ((paragraph+,
                          section*) | section+))      >
<!ELEMENT section    - - (title, paragraph+)          >
<!ELEMENT (title | paragraph)
                     - - (#PCDATA)                    >
```

The complete thesis DTD is shown in Appendix B. Appendix C shows the meaning of reference concrete syntax glyph delimiters used to indicate frequency and sequence in an element content model.

2.1.2.2. The Attribute Declaration

Attributes are a way of adding information to an element that describes certain meta-features. That is, the information does not necessarily belong in the text of the document. Attribute information is often used during document processing to achieve a means. For example, a bulleted list and a numbered list have similar structures, but require different formatting. Two different elements can be declared for each kind of list, but an attribute type declared for one list element can imply one or the other. Attributes are also useful in providing unique identifiers for an element. This is vital for installing any kind of cross-referencing in a document.

Following is an example of attributes used to distinguish between a bulleted list and a numbered list:

```
<!ELEMENT list       - - (item)+                      >
<!ATTLIST list       type (bulleted | numbered)       >
```

2.1.2.3. The Entity Declaration

An entity is a replacement for other data. There are two types: general entities and parameter entities. Of the general entities, the most common are parameter literals and external entity specifications.

A parameter literal entity is used to insert data which may be replaced during processing. For example, special characters such as an em dash might be declared in the DTD. In the SGML document, an em dash would be expressed by the em dash entity name. The processing system then decides how to handle the em dash. This is a necessary feature for interchanging documents across different systems or output instances which may handle non-ASCII characters differently.

An example of a parameter literal entity declaration follows:

```
<!ENTITY sgml "Standard Generalized Markup Language">
```

In the document instance, the full text: "Standard Generalized Markup Language" can be replaced by the entity reference: &sgml;. The reference will be replaced during processing of the final output.

External entities serve as pointers for the system to find data not stored within the document instance. They help to maintain documents in small pieces that are often easier to edit and manage than one large file. External entities are also one way of integrating non-text objects such as figures within an SGML document.

2.1.3. The Document Instance

The SGML document instance is the actual document marked up with the structural information defined in the DTD. The act of inserting these markup codes is called *tagging.* Document elements are enclosed in open and close tags bearing the name of the element as defined in the DTD. The reference concrete syntax reserves special delimiter glyphs which allows the machine and the human to recognize these open and close tags.

The document instance is pure ASCII. It and the DTD, the SGML declaration, and any external entity files comprise a complete SGML document which can be interchanged to any platform. The final output and distribution of the document is controlled by the implementor and the output engine. Because of this, the SGML document can cross platforms and conform to almost all output specifications.

Expressing the SGML document instance for final output and delivery must involve some kind of conversion. Translating an SGML document to some other format is commonly known as a down-translate. Any programming language which can perform text translation can be used to write programs for final output, but not all are suited for down-translating SGML. There are some specialized fourth generation programming languages such as Exoterica's OmniMark[2] which "chunks" or processes the SGML document instance by elements. That is, it understands the hierarchy in a document, so that writing down-translates is greatly simplified. Additionally, the process is regulated by a built-in parser so that the programmer cannot make a mistake. PERL and SED are examples of

programming languages in the public domain that can manipulate text, but they are not "SGML aware."

In some cases, delivery tools accept or "eat" raw SGML files with minimal or no conversion required. Electronic document browsers such as EBT's[3] DynaText and SoftQuad's[4] Panorama are examples of how raw SGML can be used for final output.

2.2. Validation

A good SGML based publishing system is not complete without a validating parser. A parser checks the logic and dynamics of all three components of the SGML document and alarms the user to all errors. There is no point in supporting SGML if the rules set forth by the SGML declaration and DTD are going to be ignored. Every good SGML editing and authoring software should have a parser built in[5]. There are a few SGML parsers in the public domain such as James Clark's NSGMLS[6].

2.3. Literature Review

There are not many tradebooks regarding SGML, but at the time of this thesis research, new SGML books are currently being written or published. On the other hand, on-line resources for information about SGML are very rich. The on-line SGML community is also active and growing. *Comp.text.sgml* is a newsgroup that serves as a forum for SGML discussion.

Below are brief reviews of the text materials used during research for this thesis project. Besides the technical information gained from these texts, narratives of historical case studies of SGML implementations were most useful.

The SGML Handbook by Charles Goldfarb is the complete text of ISO 8879 with the author's annotations. It is useful as a reference manual for the standard. A unique system of hypertext "buttons" and bookmarks enhances navigation and information gathering throughout the text. ISO character sets are also listed and described.

SGML: An Author's Guide to the Standard Generalized Markup Language by Martin Bryan is aimed at introducing SGML to authors. It provides guidance in using SGML to produce electronic manuscripts. It is unlikely that an author requires the breadth of SGML knowledge expressed in the book, but it is useful in expressing the importance of structure analysis.

Practical SGML by Eric van Herwijnen is an introductory book for the SGML novice. It is replete with exercises and guides the reader at an easy and well marked pace. An electronic version published using EBT's DynaText is also available.

SGML and Related Standards by Joan Smith details the benefits of having a well implemented SGML system. Examples of where SGML is being used now and the history of the standard are included. A discussion of related standards also helps to put SGML into perspective as a publishing solution.

The SGML Implementation Guide by Brian Travis and Dale Waldt is the first tradebook to so carefully detail the requirements of an SGML implementation. The business case for implementation is stressed, making it a useful book for decision making at a high managerial level. Several case studies are included to illustrate the issues raised in the book. The authoring and publishing of the book itself was a case study for the book, in that the book was produced in SGML and published in Corel Ventura for print and to HTML for the Web.

One graduate thesis was reviewed during the research for this thesis project. *SGML Based Publishing* by Erinne Cheney of the Rochester Institute of Technology is a clear survey of the SGML standard, with an emphasis on the potential benefits of such a system.

SGML is the topic of discussion in several periodical articles. A newsletter dedicated to following the standard is TAG, published by the Graphics Communication Association.

Chapter 2 Endnotes

[1] Near & Far Lite is a free software used to view DTDs in a graphical tree. It can be downloaded from the Microstar web site at: http://www.microstar.com.

[2] To learn more about Exoterica's OmniMark programming language, visit their web site at: http://www.exoterica.com.

[3] Electronic Book Technology's (EBT) web site is located at: http://www.ebt.com.

[4] SoftQuad's web site is located at: http://www.sq.com.

[5] See the SGML Editor checklist in *Practical SGML* by Eric van Herwijnen for a list of recommended features of an SGML editing tool.

[6] NSGMLS is a free SGML parser created and maintained by James Clark. It is available via FTP or at Clark's web site at: http://www.james.com. For information about available parsers and other SGML topics, visit Robin Cover's *SGML Web Page* located at: http://www.sil.org/sgml/sgml.htm.

Chapter 3

Implementation Benefits and Issues

Enhanced productivity in a publishing system is the goal of an SGML implementation. At the core of the benefits of SGML is the generic, or descriptive markup.

3.1. Specific Markup Vs. Descriptive Markup

Markup is any information in a document besides the actual content. Traditional markup are hand written instructions added to a manuscript to be prepared by a typesetter in preparation for print. The markup may include information about layout, fonts, spacing, indentation, and so on.

Electronic desktop publishing and the ubiquity of graphical user interfaces has spawned an array of software which allows the author to typeset a document in a "What You See Is What You Get" (WYSIWYG) fashion. That is, the author manipulates the final printed appearance on screen. The output engine would no be able to capture the format if not for the *specific markup* in the document, which is usually hidden from the user. Specific markup codes are usually unique to different software and systems. This is often a poor

way of creating and storing documents which must traverse platform or reach the end user in different formats.

Descriptive markup is the core of SGML. It is also known as structural or generic markup. Descriptive markup does not imply the appearance of a document, but its structure. It describes logical components of a document such as chapter, quotation, or paragraph. A generically marked up document is easily translated to whatever output instance is required. It is not appropriate for presentation but is ideal as a storage and interchange format.

SGML is a meta-language which sets forth the rules for writing a descriptive language unique to each document set. Since it is also an ISO standard, it is a guarantee of a document's usefulness for many years. On the other hand, de facto standards for page description and formatting such as Postscript or Rich Text Format (RTF) are not guaranteed to stay around. Nor are the aforementioned languages flexible enough to provide full descriptive coding.

3.2. The Benefits of SGML Based Publishing

There are many benefits of maintaining documents in a standard generic structure:

- gives documents multiple output options;
- gives document interchangeability across platforms;
- easily create searchable databases;

- easily revise lengthy, or numerous documents;
- exploit the full lifetime of a document;
- easily retrieve information from a document;

Although these benefits stem from the fact that SGML separates format from structure, this is not really a benefit unto itself. In fact, SGML documents are hard to read (although the standard was designed to make SGML readable by machines and humans). The reason they are hard to read is precisely because they lack formatting. The content of a document often relies on formatting to convey meaning. SGML documents are not intended as a formatted output medium. It is an intermediate storage format for interchange.

SGML documents are conducive to multiple output options. Programs can be written to translate the generic markup into many different formats such as Rich Text Format (RTF), TeX, HTML, etc. The emergence of new media such as the World Wide Web and CD-ROM as distribution methods for publishers has raised awareness about SGML and the importance of structural markup versus descriptive markup. Because an SGML document is pure ASCII, it can be transported to any platform. It is up to the implementor to decide how to process the information.

The structural markup in an SGML document can be used to provide unique attributes for a document piece. This enables easy editing and modifications at a future date. Also, common document features such as Tables of Contents, Indices, Cross-references, numbering and pagination can be automated and tailored for output. Text which is re-used

over and over again can be maintained as separate files and included in a document as an entity reference. This kind of text is commonly referred to as "boilerplate."

The generic structure and unique identifiers also make SGML documents easily used in a searchable database, and makes information retrieval a relatively simple affair. All of these benefits are crucial for documents which are massive and require timely re-publication. Changes to a document or document set can be made globally. Many editing operations can be automated to a certain extent by batch programming.

3.3. The Implementation Issues

The benefits discussed above come at a high cost during implementation. Accordingly, the benefits accrue as the migration from an older publishing system to a full SGML based system is completed. Implementation steps usually involve:

- cost-benefit analysis;
- document analysis;
- tools assessment, testing and acquisition;
- training of authors, editors, publishers;
- maintenance.

These benefits will be discussed and clarified in the next chapter.

Chapter 4

Methodology of This Investigation

This chapter will clarify the implementation steps stated in the previous chapter in reference to the graduate thesis publication. The graduate thesis is a publication with one author. It follows the general format of a typical research paper. A standard style manual issued by the *School of Printing Management and Sciences* sets forth the rules for the basic format and presentation of the thesis. The finished and approved thesis is bound and archived in the library. All other distribution options depend on the author's wishes. It is the responsibility of the author to procure any methods to publish the work. All typographical and editorial considerations are his under the obligations stated in the style manual.[1]

Most students in the School of Printing Management and Sciences are computer literate so few will use typewriters or outside typesetting services to publish their theses. The assumption of this case study will be made that all authors subscribe to the current desktop electronic publishing model, i.e., WYSIWYG publishing.

4.1. Cost-benefit Analysis

The first step in this case study is to perform a cost-benefit analysis. Because the graduate thesis publication is not a profit oriented system, it will suffice to say that there is no business case for an SGML implementation, unless the school intends to re-publish the works for sale at a later date. For the purposes of this thesis project, the author has injected the current model of the graduate thesis publication with hypothetical factors which do not exist in reality. Also, certain assumptions will be made.

Output requirements will be expanded to include RTF for print, HTML for the Web, and a proprietary format for inclusion in a searchable electronic browser for publication on CD-ROM. The assumption will be made that an implementor with minimal programming skills exists to maintain the SGML based publishing system. Also, all students will be assumed to have access to Microsoft Word 6.0 and to possess basic word processing skills.

The current workflow model of the graduate thesis publication is illustrated in Figure 1.

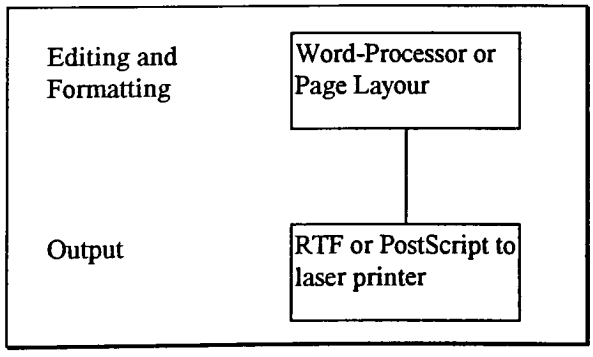

Figure 1. Current model of a Graduate Thesis Publication

Under the current desktop publishing model, the author is responsible for formatting the document according to the styles recommended in the thesis style manual. This is no trivial matter concerning the amount of time required to get the document ready for print. It is the student's responsibility to layout, print, photocopy and have bound the final thesis.

If the thesis were required to be published on the World Wide Web and for CD-ROM, the process of formatting would have to be repeated for each output instance. Additional information may be required to suit different media as well.

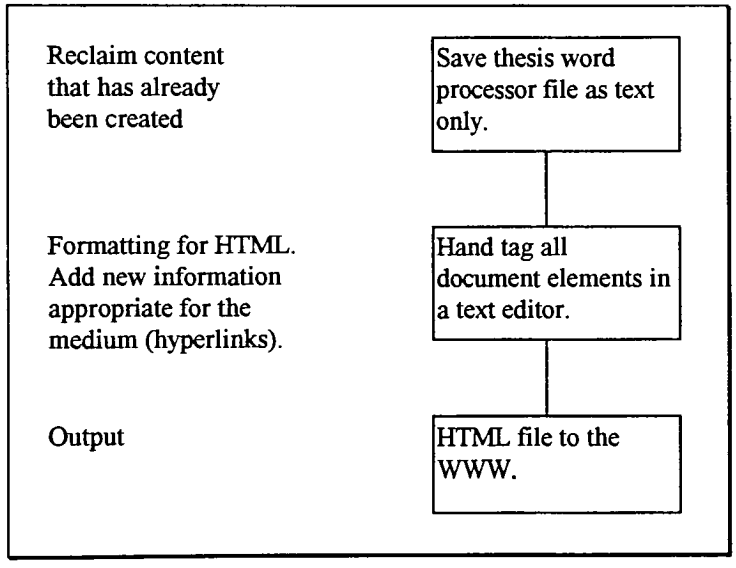

Figure 2. Model for Web publishing

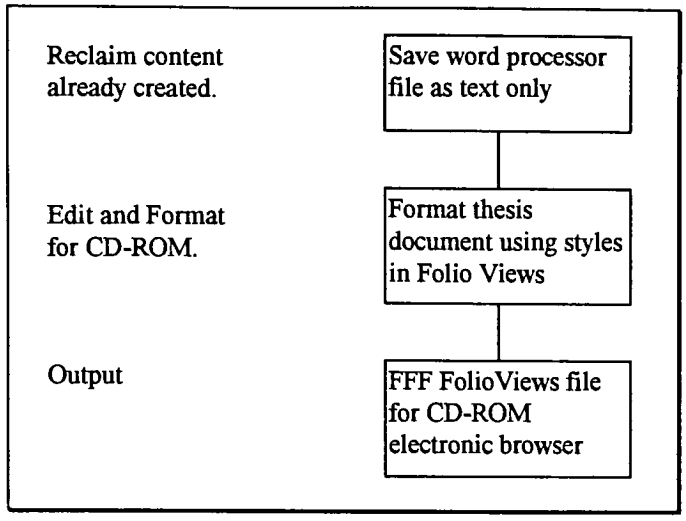

Figure 3. Model for Publishing to CD-ROM

Creating the document once for print and then having to re-format the document twice more for different media would be an unbearable time strain. The most time-consuming tasks are described below.

4.1.1. Front Matter

The front matter consists of much text which must be keyed and formatted by hand such as the Certificate of Approval, Copyright Page, Title page, Permission to Reproduce Thesis, Table of Contents, List of Tables, List of Figures, and Glossary (List of Symbols, Abbreviations, or Nomenclature).

4.1.2. Pagination

The author must paginate the thesis to the specifications in the style manual. Front matter is not numbered in the same fashion as the body and back matter. Special attention must

be paid towards widows, orphans, and maintaining continuity (i.e., section headings with first paragraphs, figure and table captions with corresponding figures and tables).

4.1.3. Applying General Styles

Applying styles in general for headings, bibliographic references, endnote references, in-line text, tables, etc., is time consuming.

4.1.4. Tables

Although tables are easy to create in MS Word 6.0, re-formatting tabular data for other formats can be very tedious.

4.1.5. Figures

Positioning figures and captions is a formatting task which can usually be handled automatically by an SGML based publishing system. It is easiest to standardize on the method of putting all figures sequentially in the back matter and using call-outs or references in the text. Very exact positioning of graphics in-line in the text is more difficult and sometimes easier to do manually instead of programmatically.

4.2. An SGML Based Publishing System

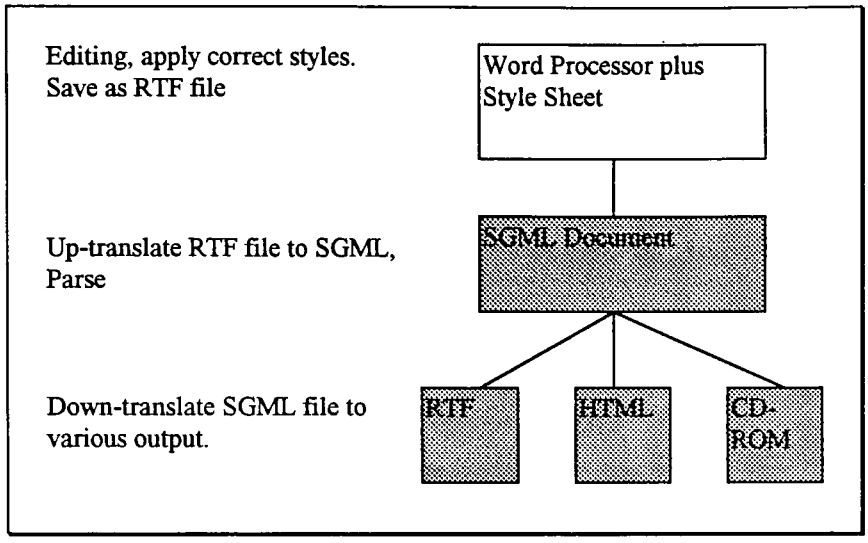

Figure 4. A Model of the Implementation of SGML Based Publishing

In Figure 4 shown above, the shaded boxes indicate processes which are totally automated through the work of one implementor and need not involve the efforts of the student. The steps required to output three different instances can be completed from the creation of one SGML file. This is in contrast to the current desktop publishing model in which the student himself would have to complete three separate, time-consuming formatting instances.

In addition, many steps towards final publication can be simplified during the editorial phase of content creation. The benefits are described below.

4.2.1. Conformance to Department Mandated Styles

A thesis submitted in valid SGML conforming to the thesis DTD is guaranteed to be processed according to the strictest style rules. This can be a benefit for the author because he does not have to waste time formatting his document. It is a benefit for the administrators because they can fully control the presentation of the graduate thesis.

4.2.2. Boilerplate Text

During the down-translation of an SGML document into its final format, valuable information can be inserted such as the boilerplate text of the front matter including the *Certificate of Approval*, *Copyright page*, *Title page*, and *the Permission to Reproduce Thesis* page. These pages are virtually identical from thesis to thesis with the exception of variables such as author name, title, date, etc. The author does not need to create these pages. Tables of Contents, Lists of Tables, Figures, and terminology can also be auto-generated.

4.2.3. Headings and Numberings

In an SGML based publishing system, the author does not need to be concerned with the numbering of chapters, sections, appendices, figures, tables, and so on. As long as these document elements are properly tagged, numbering can be handled during the down-translation of SGML to some final format. Besides saving the author time, the document reflects more accurate numbering and becomes easier to handle and edit in the future.

4.2.4. Pagination

The author does not need to paginate the final document, this is also handled during the down-translate for each different output instance.

4.2.5. Cross-referencing

An SGML based publishing system can automate the insertion of all kinds of cross references whether they are simple text messages such as "See Section X.X.X.," or whether they are actual "hotlinks" in an electronic document. The cross-references need be constructed only once by the author and not multiple times for every different output instance.

4.3. Document Analysis

The document analysis requires knowledge of the current publishing environment, and the goals of the published document. Document structure must be dissected and re-assembled in the DTD. The DTD must permit the expression of all the required information of the published document.

The structure of the thesis is a standard document structure for most research papers. The front matter is composed of a title page, permission to reproduce page, tables of contents, lists of figures, tables, terminology, acknowledgments, dedications, and an abstract. Of these, the title page, tables of contents, lists of figures, tables, and terminology can be auto-generated. Front matter which will be unique to each thesis are the author's name,

thesis title, date of publication, thesis advisor, and permission to reproduce status. These are captured in attributes.

Acknowledgments, dedications, and abstracts are elements which contain paragraphs. They are uniquely defined in the DTD because they are treated differently than the text within the body of the thesis.

Part of the document analysis is in addressing all possible uses of the information. Keeping the internet in mind as an additional output path, the thesis DTD should define elements or attributes such as email address, internet address, and keywords. Publishing in a hypertext medium such as the Web necessitates the insertion of more detailed cross-referencing. An empty element called "xref" is defined in the DTD with a "type" attribute to specify the kind of link desired. Commonly referenced document elements such as sections, figures, tables, etc., must also be given unique ID attributes so that they can be referenced.

The body matter is structured logically into chapters, sections, and subsections. Within the text of the document, one may find lists, emphasized terms (bold, italic, underlined, short quote), endnotes, tables, figures, and quotes. The introduction of a "term" element with an attribute "definition" feeds the auto-generation of terminology lists which do not need to created by the author.

The back matter consists of appendices, figures, vita, and a bibliography. The outcome of the document analysis is a DTD as shown in Appendix B.

4.4. Tools and Training

An SGML implementation will include a wide variety of tools to produce and process SGML documents. These are discussed here, along with proposed tools and methodology for the graduate thesis.

4.4.1. Authoring and Editing Tools

All one needs to produce SGML is a TEXT editor. And it is a fair assumption to say that all students have access to a computer with ASCII text editing and storing capabilities. The problem with relying on text editors alone is in training the author to understand a DTD and to properly tag their documents as they create content. This is not a viable option for most authoring environments, including the hypothetical implementation of SGML based publishing for a graduate thesis. In a mission-critical production-oriented publishing environment which could economically support the learning curve, forced learning of SGML could be a viable option.

Therefore, most applications will benefit from the use of commercial products. SGML authoring and editing software shield the author from the intricacies of structural markup and prohibit the creation of invalid SGML. The obvious advantage is that the authors are creating rigid SGML documents so no up-translate is required.

The software available today allows on-screen formatting to enhance interactive content creation. The view of SGML tags can usually be turned on or off depending on the

author's wishes. Attribute information can be edited using dialog boxes. SGML authoring software often integrates the tools common to a word processor such as spell checkers and search and replace mechanisms. These factors can help ease the implementation issue of training author and editors to write SGML.

However, training and support is still required, as the introduction of any new software package would. Although the current and projected demand for sophisticated SGML editing tools is increasing, these packages are expensive. Therefore, the use of SGML editing tools is not a viable option for the implementation of SGML based publishing for a graduate thesis. Cost and limited support would severely decrease participation.

Neither is hand-tagging an SGML document after it has been created a viable option for this hypothetical implementation. However, it is worth mentioning because it is very common in reality. The cost of hand keying text is very low and for profitable industries with much legacy data, it is an economical alternative to authoring SGML.

4.4.1.1. Word Processors and Style Sheets

An assumption was made earlier in this investigation, that all authors in the graduate electronic publishing program will have access to Microsoft Word 6.0, and that they are computer literate and competent in the use of this software. This assumption is made to introduce a proposed method of acquiring SGML by using word processor style sheets.

Style sheets or templates are often used as tools for regulating house style. It is a way for authors to produce compliant documents without the use of written style manuals and explicit instructions. Style sheets also offer the benefit of providing a "flat" descriptive kind of markup similar to SGML, in that the style names provide structural information. The descriptive style names generated in RTF provides a manner of up-translating the RTF file to SGML.

RTF is a proprietary text format promulgated by Microsoft and is supported by many other software manufacturers today. It is a convenient method of transporting documents to different word processors and platforms, but only if RTF is supported. An up-translate can be written using a text conversion language or by using a conversion tool such as EBT's Dynatag[2] which maps RTF styles to SGML elements rules. The product of this kind of "pre-packaged" conversion tool is not intended to be the final, valid SGML instance, but an intermediary format. This intermediate format will require further translation by hand or by a translating program to clean it up. The reason for this is that RTF styles are "flat." That is, they do not allow for hierarchical structure. They also do not record attribute information.

Text conversion often involves a great deal of simple find-and-replace commands which require very little thought or time. However, there is a smaller percentage of text elements which require a lot of thought and wily programming skills, or manual intervention. This is known as the "80/20" rule[3]. That is, eighty percent of the document requires twenty

percent of the programming time, and twenty percent of the document requires the other eighty percent.

For the hypothetical implementation of SGML based publishing for a graduate thesis, the use of a standardized style sheet is the best choice for acquiring SGML documents. A Microsoft Word document template is small in size and easy to distribute. Access to this software is available to most students. Finally, the author does not need to be trained in SGML, HTML, or any other format. All these factors would maximize the success of an implementation of this kind.

Relying on word processors and style sheets is not without its shortcomings. A word processor cannot be used to produce a "clean" or "valid" document. Validity will depend on the proper application of the pre-defined style names by the author. Certain measures may be taken to regulate use of the style sheet. The MS Word interface may be customized using a dialog editor to minimize the chance of erroneous actions such as defining new style names.

Also, word processors cannot edit meta-data about text in the document. SGML editing software would provide dialog boxes for attribute information. In a word processor, attribute information such as IDs, Author Name, Advisor Name, etc., must be keyed in-line with the real document text as markup. Enforcing proper keying sequences is a difficult step to expect all authors to willingly adopt.

Another potential problem is that of providing style sheets for all the word processors used by students. Although in a hypothetical situation, the choice of Microsoft Word can be mandated, in reality, students are using other versions of MS Word, or other manufacturer's word processors. Some dated, or less sophisticated word processors do not support in-line character styles definitions, which poses difficulty for expressing in-line elements such as short quotations, terms, definitions, etc. Since all word processors do not behave the same or produce the same RTF for that matter, slightly different training procedures and text conversions would be required.

4.4.2. Text conversion

As discussed previously, there are many commercial and free programming languages which can perform text manipulations. For the conversion of SGML to other formats, an "SGML aware" language such as OmniMark is preferred over "non-SGML aware" languages. For this investigation, a free demonstration version of OmniMark was used to write small programs to illustrate two simple down-translates of SGML to RTF and HTML. The programs and output files can be seen in Appendix D. A license for a fully functional registered copy is costly, but only one license is required in this implementation because authors would not perform any programming or conversion.

4.4.3. Training

Training authors in proper use of a style sheet would be required in an SGML implementation. Recognizing all the document elements and style names is key, especially

for attribute information. The style manual would be completely re-written. Formatting instruction would not be provided, but must include specific information on the semantics of style names, and how to properly introduce attribute information. Information on how to integrate figures and tables would also be necessary.

An implementor is also required to administer an SGML based publishing system for a graduate thesis. This person must possess knowledge of SGML, HTML, RTF, and the ability to learn any new formats as required. The implementor also must write and maintain a DTD and translating programs. Style sheets for all adopted word processors must be created and technical support must be provided to authors.

Chapter 4 Endnotes

[1] The Master of Science Degree Thesis Style Manual for Graduate Programs in the RIT School of Printing Management and Sciences is available through the Program director at this institution.

[2] For a complete listing of conversion tools, see the SGML Web page at: http://www.sil.org/sgml/sgml.htm.

[3] SGML '95 Conference in Boston, MA, December 1995.

Chapter 5

Conclusions

This thesis project sought to clarify the requirements of an implementation of SGML based publishing while investigating the relative benefits of such a system. The publication of a graduate thesis in the RIT School of Printing Management and Sciences was used as an example. The current model of desktop publishing was illustrated in the previous chapter while pointing out the inherent problems if HTML output and CD-ROM output were required along with formatted print output.

In the hypothetical implementation of SGML based publishing of a graduate thesis publication, a set of tools was proposed. A Microsoft Word 6.0 template would be created and distributed. This style sheet would be used by the student to author and edit his thesis document. The content need not be formatted, but the pre-defined styles must be used correctly. The author would then submit the finished thesis content as an RTF file along with any external files such as graphics. The student's interaction with the publishing process would end there.

The RTF file would then be translated to SGML using a program written in OmniMark. The SGML file is ready for final output when the file parses according to the thesis DTD.

An unparseable file would most likely have to be fixed manually. Down-translate programs will then convert the SGML file to a print-ready RTF file, HTML file, and to a proprietary searchable browser for CD-ROM.

Although writing good translating programs is a lengthy and often expensive task, they need only be written once for the up-translate and for each down-translate. And in many cases, code can be re-used by the smart programmer. the student only creates one document instance of his thesis and almost instantaneously three different outputs can be generated with no formatting required by the student.

With the available tools, a brief sample thesis document was hand-tagged in SGML with a test DTD (this removes the need of an up-translate step). Two down translate programs were written using the OmniMark language to convert the sample thesis to RTF and HTML. These can be seen in Appendix D.

In conclusion, the benefits of implementing an SGML based publishing system for a graduate thesis is valid only under the hypothetical circumstances of forced output to additional formats. The benefit are:

- Conformance to house styles;
- Content only created once;
- Output simultaneously to any output format;
- Student does not have to worry about formatting;
- Student does not have to key boilerplate text;

Even then, the implementation costs are high. The biggest problem is the lack of an implementor. Graduate theses are currently published completely by the students themselves. However, in a profit-driven publishing environment, SGML is a clear solution for many of the publishing problems present in the age of electronic publishing.

Bibliography

Bibliography

Bryan, Martin, (1988). *SGML: An Author's Guide to the Standard Generalized Markup Language*. Cheltanham, UK: Addison Wesley Publishing Co., Inc.

Cheney, Erinne, (1992). *SGML Based Publishing*. M.S. Degree, The Rochester Institute of Technology.

Cover, Robin, (1995). *The SGML Web Page*. http://www.sil.org/sgml/sgml.htm

Exoterica Corporation. (1995). *OmniMark Manuals*. http://www.exoterica.com

Goldfarb, Charles. (1990). *The SGML Handbook*. Yuri Rubinsky (Ed.) Oxford: Clarendon Press.

Ressler, Sandy, (1993). *Perspectives on Electronic Publishing*. Englewood Cliffs, NJ: PTR Prentice-Hall Limited.

Smith, Joan M., (1992). *SGML and Related Standards*. Chichester, West Sussex, England: Ellis Horwood Limited.

Travis, Brian & Waldt, Dale, (1995). *The SGML Implementation Guide*. New York: Springer-Verlag.

van Herwijnen, Eric, (1994). *Practical SGML, Second Ed.* Boston: Kluwer Academic Publishers.

Appendices

Appendix A

Appendix A

SGML Declaration

```
<!SGML "ISO 8879:1986"
                                CHARSET
BASESET "ISO 646-1983//CHARSET International Reference
Version (IRV)//ESC 2/5 4/0"
DESCSET 0 9 UNUSED
9 2 9
11 2 UNUSED
13 1 13
14 18 UNUSED
32 95 32
127 1 UNUSED
BASESET "ISO Registration Number 109//CHARSET ECMA-94 Right
Part of Latin Alphabet Nr. 3//ESC 2/5 4/0" DESCSET 128 32
UNUSED 160 5 32 165 1 UNUSED 166 88 38 254 1 127 255 1
UNUSED
CAPACITY SGMLREF TOTALCAP 175000 ENTCAP 50000 ENTCHCAP 50000
ELEMCAP 50000 GRPCAP 70000 EXGRPCAP 50000 EXNMCAP 50000
ATTCAP 50000 ATTCHCAP 50000 AVGRPCAP 50000 NOTCAP 50000
NOTCHCAP 50000 IDCAP 50000 IDREFCAP 50000 MAPCAP 50000
LKSETCAP 50000 LKNMCAP 50000 SCOPE DOCUMENT SYNTAX
SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127 255 BASESET
"ISO 646-1983//CHARSET International Reference Version
(IRV)//ESC 2/5 4/0"
DESCSET 0 128 0
FUNCTION RE 13 RS 10 SPACE 32 TAB SEPCHAR 9 NAMING LCNMSTRT
"" UCNMSTRT "" LCNMCHAR "-." UCNMCHAR "-." NAMECASE GENERAL
YES ENTITY
NO DELIM GENERAL SGMLREF SHORTREF NONE NAMES SGMLREF
QUANTITY SGMLREF LITLEN 2048 NAMELEN 32 ATTCNT 80 FEATURES
MINIMIZE DATATAG NO OMITTAG YES RANK NO SHORTTAG YES LINK
SIMPLE NO IMPLICIT NO EXPLICIT NO OTHER CONCUR NO SUBDOC NO
FORMAL YES APPINFO NONE >
```

Appendix B

Appendix B

Thesis Document Type Definition

```
<!-- DTD for the graduate thesis in the RIT School of
Printing Management and Sciences.                          -->

<!DOCTYPE thesis  [

<!ENTITY  % char1 PUBLIC "ISO 8879-1986//ENTITIES
Publishing//EN" "\entities\iso-pub.ent"                    >
%char1;

<!ELEMENT thesis        - - (front, body, back)
                             + (footnote | term |
                                emphasis | xref)           >
<!ATTLIST thesis        author          CDATA   #REQUIRED
                        month           CDATA   #REQUIRED
                        year            CDATA   #REQUIRED
                        advisor         CDATA   #REQUIRED
                        title           CDATA   #REQUIRED
                        permission (grant | notify | deny)
                                                #REQUIRED
                        email           CDATA   #REQUIRED
                        internet        CDATA   #REQUIRED
                        keywords        CDATA   #REQUIRED  >

<!-- ************ INCLUSION ELEMENTS *****************  -->
<!-- These elements are allowed anywhere in the document -->

<!ELEMENT footnote      - - (#PCDATA)    -(footnote)       >
<!ATTLIST footnote      id              CDATA   #REQUIRED  >

<!ELEMENT term          - - (#PCDATA)                      >
<!ATTLIST term          def             CDATA   #REQUIRED  >

<!ELEMENT emphasis      - - (#PCDATA)                      >
<!ATTLIST emphasis      type (bold | italic | underline |
                              shortquote | longquote | code)
```

```
                                                    #REQUIRED  >

<!ELEMENT xref           - - EMPTY                                >
<!ATTLIST XREF           id                IDREF   #REQUIRED
                         type (footnote | figure | table |
                               chapter | section | http |
                               email | glossary) #REQUIRED  >

<!ENTITY % paratext "para | list | figure | table"               >

<!-- ********* PARATEXT ELEMENTS ********************** -->

<!ELEMENT para           - O (#PCDATA)                            >

<!ELEMENT list           - O (item+)                              >
<!ATTLIST list           type (bullet | number)  bullet           >

<!ELEMENT item           - O (#PCDATA)                            >

<!ELEMENT figure         - - (title, graphic)                     >
<!ATTLIST figure         id                ID      #REQUIRED  >

<!ELEMENT title          - - (#PCDATA)                            >

<!ELEMENT graphic        - O EMPTY                                >
<!ATTLIST graphic        filename          CDATA   #REQUIRED
                         width             CDATA   #REQUIRED
                         height            CDATA   #REQUIRED  >

<!ELEMENT table          - - (title, tablehead?, tablebody) >
<!ATTLIST table          id                ID      #REQUIRED  >

<!ELEMENT (tablehead | tablebody)
                         - - (row)+                               >

<!ELEMENT row            - - (cell)+                              >
<!ELEMENT cell           - - (#PCDATA)                            >

<!-- ********** FRONT MATTER ************************** -->

<!ELEMENT front          - O (dedication?, acknowledgments?,
                              toc, lot?, lof?, glossary?,
                              abstract)                           >

<!ELEMENT dedication     - O (para)                               >
```

```
<!ELEMENT acknowledgments
                        - O (para)                              >

<!-- The Table of Contents (TOC), List of Tables (LOT),
     List of Figures (LOF), and the Glossary are
     auto-generated                                             -->

<!ELEMENT (toc | lot | lof | glossary)
                        - O EMPTY                               >

<!ELEMENT abstract      - O (%paratext;)+       -(figure)       >

<!-- *************** BODY MATTER ********************** -->

<!ELEMENT body          - O (chapter)+                          >

<!ELEMENT chapter       - O (title, (((%paratext;)+,
                             section*) | section*) )            >
<!ATTLIST chapter       id              ID      #REQUIRED       >

<!ELEMENT section       - - (title, (((%paratext;)+,
                             section | section*) )              >
<!ATTLIST section       id              ID      #REQUIRED       >

<!-- ********** BACK MATTER ***************************** -->

<!ELEMENT back          - O (bibliography, appendix*,
                             vita?)                             >

<!ELEMENT bibliography  - O (citation)+                         >
<!ELEMENT citation      - O (#PCDATA)                           >

<!ELEMENT appendix      - O (%paratext;)+                       >
<!ATTLIST appendix      id              ID      #REQUIRED       >

<!ELEMENT vita          - O (para)+                             >

]>
```

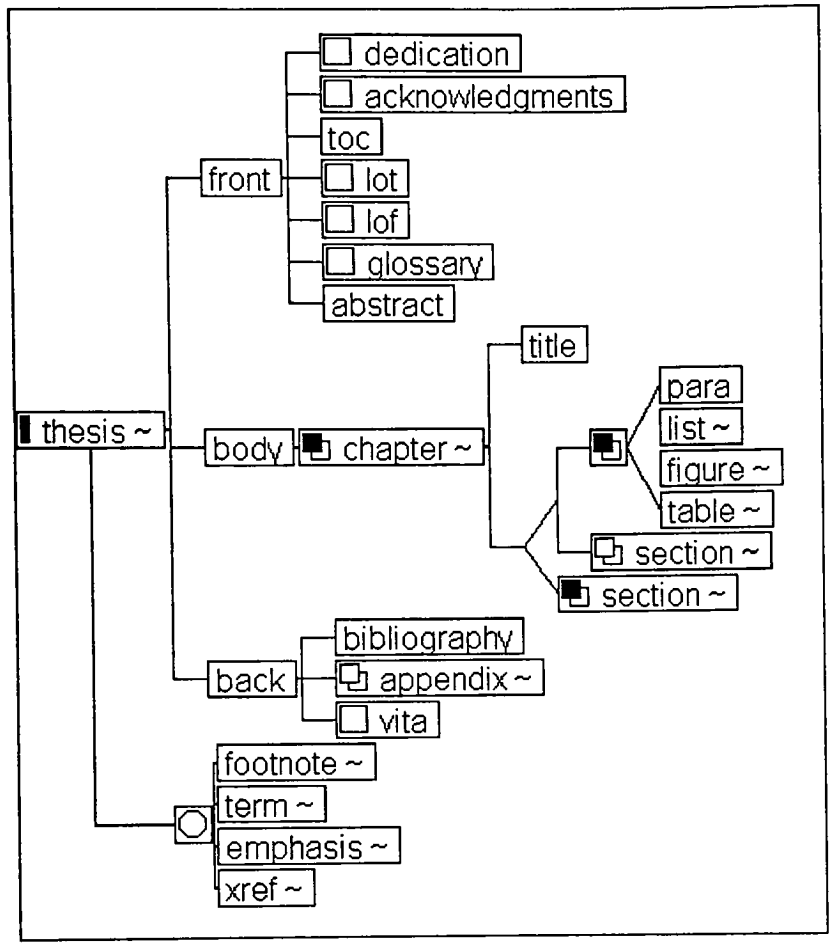

Figure 5. Thesis DTD Tree Diagram

Appendix C

Appendix C

Frequency and Occurrence Indicators

Frequency Indicators:

?	Zero or None allowed
*	Zero or More allowed
+	One or More allowed

Occurrence Indicators:

,	Followed by
|	One or the other; in any order
&	Both in any Order

Appendix D

Appendix D

Down-translates

Input File: test.dtd (SGML declaration in DTD), abstract.sgm

```
<!DOCTYPE thesis [

<!ELEMENT thesis        - - (abstract)                          >
<!ATTLIST thesis        author          CDATA   #REQUIRED
                        month           CDATA   #REQUIRED
                        year            CDATA   #REQUIRED
                        advisor         CDATA   #REQUIRED
                        title           CDATA   #REQUIRED
                        permission (grant | notify | deny)
                                                #REQUIRED
                        email           CDATA   #REQUIRED
                        internet        CDATA   #REQUIRED  >

<!ELEMENT abstract      - - (para)+                             >
<!ELEMENT para          - O (#PCDATA)                           >

]>

<THESIS AUTHOR="Jin Park" MONTH="February" YEAR="1996"
        ADVISOR="Professor Frank Cost" TITLE="The
        Investigation of an Implementation of SGML Based
        Publishing of a Graduate Thesis" PERMISSION="grant"
        EMAIL="jyp5868@grace.rit.edu"
        INTERNET="http://www.rit.edu/~jyp5868">

<ABSTRACT>
<PARA>The Standard Generalized Markup Language (SGML) has
been the International Organization of Standardization (ISO)
published standard for text interchange for nearly a decade.
Since 1986, SGML based publishing has been successfully
implemented in many fields, notably those industries with
```

massive and mission-critical publishing operations such as the military, legal, medical, and heavy industries.

<PARA>SGML based publishing differs from the WYSIWYG paradigm of desktop publishing in that an SGML document contains descriptive, structural markup rather than specific formatting markup. Specific markup describes the appearance of a document and is usually a proprietary code which makes the document difficult to re-use or interchange to different systems.

<PARA>The structurally generic markup codes in an SGML document allow the fullest exploitation of the information. An SGML document exhibits more re-usability than a document created and stored in a proprietary formatting code. In many cases, workflow and production are greatly improved by the implementation of SGML based publishing.

<PARA>Historical and anecdotal case studies of many applications clearly delineate the benefits of an SGML based publishing system. And certainly, the boom in Web publishing has spurred interest in enabling a publishing system with multi-output functionality. However, implementation is associated with high costs. The acquisition of new tools and new skills is a costly investment. A careful cost-benefit analysis must determine that the current publishing needs would be satisfied by moving to SGML.
Increased productivity is the measure by which SGML is adopted.

<PARA>The purpose of this thesis is to investigate the relative benefits and requirements of a simple SGML based publishing implmentation. The graduate thesis publication of the Rochester Institute of Technology, School of Printing Management and Sciences was used as an example.

<PARA>The author has hypothetically injected the publication process of the graduate thesis with factors which do not exist in reality. The required output has been expanded from mere print output to include publishing to the World Wide Web (WWW) in the Hypertext Markup Language (HTML), and to some proprietary electronic browser such as Folio Views for inclusion in a searchable collection of graduate theses on CD-ROM. A proposed set of tools and methods are discussed in order to clarify the requirements of such as an SGML implementation.
</ABSTRACT>
</THESIS>

SGML to RTF program: mkrtf.xom

```
DOWN-TRANSLATE

element thesis

output "{\rtf\ansi\marl2160\margr1440\margt1800\margb1800
%n"
output "{\qc\ %n"
output "{\sb1080 School of Printing Management and
Sciences\par} %n"
output "{\sb0 Rochester Institute of Technology\par} %n"
output "{\sb0 Rochester, New York\par} %n"
output "{\sb540\b Certificate of Approval\par} %n"
output "{\sb396 __________\par} %n"
output "{\sb540\b Master's Thesis\par} %n"
output "{\sb396 __________\par} %n"
output "%n"
output "{\sb396 This is to certify that the Master's Thesis
of\par} %n"
output "{\sb540 %v(author)\par} %n"
output "{\sb396 With a major in Graphic Arts Publishing\par}
%n"
output "{\sb0 has been approved by the Thesis Committee as
satisfactory\par %n"
output "for the thesis requirement for the Master of Science
degree\par %n"
output "at the convocation of\par} %n"
output "{\sb240 %v(month) %v(year)\par} %n"
output "{\sb540 Thesis Committee:\par} %n"
output "{\sb240 --------------------\par} %n"
output "{\sb0\fs12 Thesis Advisor\par} %n"
output "{\sb240 --------------------\par} %n"
output "{\sb0\fs12 Graduate Program Coordinator\par} %n"
output "{\sb240 --------------------\par} %n"
output "{\sb0\fs12 Director or Designate\par} %n"
output "%n"
output "{\sb2880\li1440\ri1440\b %v(title)\par} %n"
output "{\sb1440 by\par} %n"
output "{\sb240 %v(author)\par} %n"
output "{\sb2000 A thesis project submitted in partial
fulfillment of the\par} %n"
output "{\sb0 requirements for the degree of Master of
Science in the\par %n"
output "School of Printing Management and Sciences in the
College\par %n"
output "of Imaging Arts and Sciences of the\par %n"
```

```
output "Rochester Institute of Techonology\par} %n"
output "{\sb1440 %v(month) %v(year)\par %n"
output "Thesis Advisor:  %v(advisor)\par}} %n"
output "%n"
output "{\page Title of Thesis:  {\b %v(title)}\par\par} %n"
output "{I, %v(author), hereby, {\b grant permission} %n"
output "to the Wallace Memorial Library of R.I.T. to %n"
output "reproduce my thesis in whole or in part. Any %n"
output "reproduction will not be for commercial use or %n"
output "profit.\par\par}%n"
output "{\sb240 Date:  %v(month) %v(year)\par} %n"
output " %c} %n"

element abstract
        output "{\page\sb1440\qc\b Abstract\par} %c %n"

element para
        output "%n{\sb280\s1560 %c\par}"
```

SGML to HTML Program: mkhtml.xom

```
DOWN-TRANSLATE

element thesis

output "<h1><center>%v(title)</h1>%n"
output '<br><h2>by <a href="%v(internet)">%v(author)</a>%n'
output '<br><a href="mailto:%v(email)">%v(email)</a></h2>%n'
output "<br><h3>A thesis project submitted in partial %n"
output "fulfillment of the requirements for the degree %n"
output "of Master of Science in the School of Printing %n"
output "Management and Sciences in the College of Imaging
%n"
output "Arts and Sciences of the Rochester Institute of %n"
output "Technology %n"
output "<br>%v(month) %v(year) %n"
output "<br>Thesis Advisor:  %v(advisor)%c%n%n</h3>"

element abstract
        output "<hr><h2>Abstract</h2></center> %n"
        output "<br>%c%n"

element para
        output "<p>%c"
```

RTF Output: abstract.rtf

```
{\rtf\ansi\marl2160\margr1440\margt1800\margb1800
{\qc\
{\sb1080 School of Printing Management and Sciences\par}
{\sb0 Rochester Institute of Technology\par}
{\sb0 Rochester, New York\par}
{\sb540\b Certificate of Approval\par}
{\sb396 __________\par}
{\sb540\b Master's Thesis\par}
{\sb396 __________\par}

{\sb396 This is to certify that the Master's Thesis of\par}
{\sb540 Jin Park\par}
{\sb396 With a major in Graphic Arts Publishing\par}
{\sb0 has been approved by the Thesis Committee as
satisfactory\par
for the thesis requirement for the Master of Science
degree\par
at the convocation of\par}
{\sb240 February 1996\par}
{\sb540 Thesis Committee:\par}
{\sb240 --------------------\par}
{\sb0\fs12 Thesis Advisor\par}
{\sb240 --------------------\par}
{\sb0\fs12 Graduate Program Coordinator\par}
{\sb240 --------------------\par}
{\sb0\fs12 Director or Designate\par}

{\sb2880\li1440\ri1440\b The Investigation of an
Implementation of SGML Based Publishing of a Graduate
Thesis\par}
{\sb1440 by\par}
{\sb240 Jin Park\par}
{\sb2000 A thesis project submitted in partial fulfillment
of the\par}
{\sb0 requirements for the degree of Master of Science in
the\par
School of Printing Management and Sciences in the
College\par
of Imaging Arts and Sciences of the\par
Rochester Institute of Techonology\par}
{\sb1440 February 1996\par
Thesis Advisor:  Professor Frank Cost\par}}
```

{\page Title of Thesis: {\b The Investigation of an Implementation of SGML Based Publishing of a Graduate Thesis}\par\par}
{I, Jin Park, hereby, {\b grant permission} to the Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.\par\par}
{\sb240 Date: February 1996\par}
{\page\sb1440\qc\b Abstract\par}
{\sb280\s1560 The Standard Generalized Markup Language (SGML) has been the International Organization of Standardization (ISO) published standard for text interchange for nearly a decade. Since 1986, SGML based publishing has been successfully implemented in many fields, notably those industries with massive and mission-critical publishing operations such as the military, legal, medical, and heavy industries.\par}
{\sb280\s1560 SGML based publishing differs from the WYSIWYG paradigm of desktop publishing in that an SGML document contains descriptive, structural markup rather than specific formatting markup. Specific markup describes the appearance of a document and is usually a proprietary code which makes the document difficult to re-use or interchange to different systems. \par}
{\sb280\s1560 The structurally generic markup codes in an SGML document allow the fullest exploitation of the information. An SGML document exhibits more re-usability than a document created and stored in a proprietary formatting code. In many cases, workflow and production are greatly improved by the implementation of SGML based publishing.\par}
{\sb280\s1560 Historical and anecdotal case studies of many applications clearly delineate the benefits of an SGML based publishing system. And certainly, the boom in Web publishing has spurred interest in enabling a publishing system with multi-output functionality. However, implementation is associated with high costs. The acquisition of new tools and new skills is a costly investment. A careful cost-benefit analysis must determine that the current publishing needs would be satisfied by moving to SGML.
Increased productivity is the measure by which SGML is adopted.\par}
{\sb280\s1560 The purpose of this thesis is to investigate the relative benefits and requirements of a simple SGML based publishing implmentation. The graduate thesis publication of the Rochester Institute of Technology, School

of Printing Management and Sciences was used as an example.\par}
{\sb280\s1560 The author has hypothetically injected the publication process of the graduate thesis with factors which do not exist in reality. The required output has been expanded from mere print output to include publishing to the World Wide Web (WWW) in the Hypertext Markup Language (HTML), and to some proprietary electronic browser such as Folio Views for inclusion in a searchable collection of graduate theses on CD-ROM. A proposed set of tools and methods are discussed in order to clarify the requirements of such as an SGML implementation.\par}
}

HTML Output: abstract.htm

```
<h1><center>The Investigation of an Implementation of SGML
Based Publishing of a Graduate Thesis</h1>
<br><h2>by <a href="http://www.rit.edu/~jyp5868">Jin
Park</a>
<br><a
href="mailto:jyp5868@grace.rit.edu">jyp5868@grace.rit.edu</a
></h2>
<br><h3>A thesis project submitted in partial
fulfillment of the requirements for the degree
of Master of Science in the School of Printing
Management and Sciences in the College of Imaging
Arts and Sciences of the Rochester Institute of
Technology
<br>February 1996
<br>Thesis Advisor:  Professor Frank
Cost<hr><h2>Abstract</h2></center>
<br><p>The Standard Generalized Markup Language (SGML) has
been the International Organization of Standardization (ISO)
published standard for text interchange for nearly a decade.
Since 1986, SGML based publishing has been successfully
implemented in many fields, notably those industries with
massive and mission-critical publishing operations such as
the military, legal, medical, and heavy industries.<p>SGML
based publishing differs from the WYSIWYG paradigm of
desktop publishing in that an SGML document contains
descriptive, structural markup rather than specific
formatting markup. Specific markup describes the appearance
of a document and is usually a proprietary code which makes
the document difficult to re-use or interchange to different
systems. <p>The structurally generic markup codes in an SGML
document allow the fullest exploitation of the information.
An SGML document exhibits more re-usability than a document
created and stored in a proprietary formatting code. In many
cases, workflow and production are greatly improved by the
implementation of SGML based publishing.<p>Historical and
anecdotal case studies of many applications clearly
delineate the benefits of an SGML based publishing system.
And certainly, the boom in Web publishing has spurred
interest in enabling a publishing system with multi-output
functionality. However, implementation is associated with
high costs. The acquisition of new tools and new skills is a
costly investment. A careful cost-benefit analysis must
determine that the current publishing needs would be
satisfied by moving to SGML.
```

Increased productivity is the measure by which SGML is adopted.<p>The purpose of this thesis is to investigate the relative benefits and requirements of a simple SGML based publishing implmentation. The graduate thesis publication of the Rochester Institute of Technology, School of Printing Management and Sciences was used as an example.<p>The author has hypothetically injected the publication process of the graduate thesis with factors which do not exist in reality. The required output has been expanded from mere print output to include publishing to the World Wide Web (WWW) in the Hypertext Markup Language (HTML), and to some proprietary electronic browser such as Folio Views for inclusion in a searchable collection of graduate theses on CD-ROM. A proposed set of tools and methods are discussed in order to clarify the requirements of such as an SGML implementation.

</h3>